



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет
імені Івана Пулюя

Кафедра автоматизації технологічних
процесів та виробництв

Методичні вказівки
для виконання лабораторної роботи №5
“Вивчення інтегрованого середовища програмування
Keil ”
з курсу “Розробка систем керування на основі
OMEOM”

Тернопіль 2016

Методичні вказівки для виконання лабораторної роботи №5 «Вивчення інтегрованого середовища програмування Keil» з курсу «Розробка систем керування на основі ОМЕОМ».

Методичні вказівки розглянуті і схвалені кафедрою «Автоматизація технологічних процесів і виробництв», протокол № 4 від 21.11.2016 р.

Відповідальні за випуск

доцент, к.т.н. Медвідь В.Р.,
асистент Пісьціо В.П.

Лабораторна робота №5

Вивчення інтегрованого середовища програмування Keil

1. Мета роботи

- 1.1. Вивчити інтегроване середовище програмування Keil.
- 1.2. Отримати навички роботи з текстовим редактором цього середовища програмування.
- 1.3. Отримати навички роботи з програмними проектами.
- 1.4. Навчитися транслювати програми.
- 1.5. Вивчити роботу відлагоджувача програм в інтегрованому середовищі програмування Keil.

2. Підготовка до виконання роботи

Вивчити наступні питання:

- 2.1. Вивчити порядок створення програмного проекту в інтегрованому середовищі програмування Keil.
- 2.2. Вивчити налаштування властивостей програмного проекту в інтегрованому середовищі програмування Keil.
- 2.3. Вивчити використання вікна управління програмним проектом в інтегрованому середовищі програмування Keil.
- 2.4. Вивчити методи трансляції окремих файлів в інтегрованому середовищі програмування Keil.
- 2.5. Вивчити методи трансляції програмного проекту в інтегрованому середовищі програмування Keil.
- 2.6. Вивчити роботу відлагоджувача програм в інтегрованому середовищі програмування Keil.

3. Завдання до роботи

- 3.1. Увійдіть в інтегроване середовище програмування.
 - 3.2. Створіть новий файл вихідного тексту для однієї з чотирьох програм відповідно до Вашого варіанту (таблиця 1). Файл може бути, наприклад, L1.asm (розширення **.asm** обов'язкове).
- Тексти програм подані нижче.

Програма 1. Програмування ПЗП даних Flash/EEPROM

ПЗП даних типу Flash/EEPROM призначена для зберігання оперативних даних керуючої програми в перервах між відключеннями мікроконтролера.

Для запису даних в цю пам'ять слід дотримуватися наступної послідовності операцій:

1. Записати адресу сторінки в регістри EADRL та EADRH.
2. Записати дані в регістри даних EDATA1- EDATA4.
3. Записати в регістр ECON команду.

Перед записом інформації в ПЗП даних попередньо слід стерти наявну інформацію. При цьому в байти записуються значення FFh.

Так як стирання можливе тільки для сторінки цілком, запис одного байту вимагає попереднього запам'ятовування сторінки, наприклад, в регістрах даних EDATA1-EDATA4.

Програма запису байту 1 в сторінку 2:

```
MOV EADRL, #2h      ; встановлення сторінки
MOV EADRH, #0
MOV ECON, #1        ; читання сторінки
MOV EDATA1, #33h    ; зміна байту
MOV ECON, #5        ; стирання сторінки
```

```

MOV ECON, #2          ; запис сторінки
MOV ECON, #4          ; перевірка запису
MOV A, ECON
JNZ WRITE ERROR

```

Програма 2. Програма реалізації максимальної часової затримки за допомогою таймера (32 ms)

```

D32ms:
MOV TCON, #1          ; задання режиму роботи таймера 0
MOV TL0, #0           ; очистка молодшого розряду таймера
MOV TH0, #0           ; очистка старшого розряду таймера
SETB TR0              ; запуск таймера
JNB TF0, $            ; очікування встановлення флага переповнення таймера
CLR TF0
RET

```

Програма 3. Пересилання в ЗПД (зовнішню пам'ять даних) і з ЗПД

Для пересилання даних в ЗПД і з ЗПД використовується команда MOVX та шістнадцятирозрядний регістр показчика даних DPTR.

Програма запису константи 255 в ЗПД за абсолютною адресою 000000h:

```

MOV DPP, #0
MOV DPH, #0
MOV DPL, #0
MOV A, #255
MOVBX @DPTR, A

```

Програма пересилання даних з комірки пам'яті ЗПД з адресою 20, що знаходиться на сторінці 1, в акумулятор.

```

MOV DPP, #1
MOV DPH, #0
MOV DPL, #20
MOVBX A, @DPTR

```

Програма 4. Програмування АЦП і вивід даних через порт

Програма виконує прості аналого-цифрові перетворення і виводить результати в послідовний порт UART. Запалює червоний світлодіод після завершення кожного нового перетворення, що ініціюються кожні 200 мс

; Всі розрахунки швидкості допускають 2.097152MHz MCLK.;

```

LED EQU P3.4          ; до виводу P3.4 під'єднано червоний світлодіод
CHAN EQU 0            ; присвоїти величині CHEN вхідного каналу АЦП
                        ; значення 0 через 6
                        ; початок коду

```

```

CSEG
ORG 0000h
JMP MAIN              ; перехід на програму main

```

; Програма MAIN

```

ORG 004Bh
MAIN:

```

```

                                ; налаштування UART
MOV  T3CON,#083H
MOV  T3FD,#02DH
MOV  SCON,#52h
                                ; попереднє налаштування
MOV  ADCCON1,#0ACh            ; увімкнути АЦП

; виконати одноразове просте перетворення

AGAIN: CPL  LED                ; загасити світлодіод
      MOV  A,#020              ; встановити тривалість затримки 200ms
      CALL DELAY              ; перейти на підпрограму delay

      SETB SCONV              ; ініціювати просте АЦП перетворення
                                ; і викликати переривання АЦП після

      JNB  ADCL,$

      MOV  A,ADCDATAH
      CALL SENDVAL
      MOV  A,ADCDATAL
      CALL SENDVAL

      JMP  AGAIN              ; перейти на початок програми

                                ; підпрограма затримки
DELAY:
      MOV  R1,A                ; встановити першу затримку
DLY0:  MOV  R2,#01Bh           ; встановити затримку loop0
DLY1:  MOV  R3,#0FFh          ; встановити затримку loop1
      DJNZ R3,$                ; декремент R3 і перехід, поки R3 не рівне 0
      DJNZ R2,DLY1            ; декремент R2 і перехід на DLY1, поки R2 не рівне 0
      DJNZ R1,DLY0            ; декремент R1 і перехід на DLY0, поки R1 не рівне 0
      RET                      ; повернення з підпрограми

END

```

3.3. Створіть проект з ім'ям LAB1.

3.4. Додайте в проект файл з програмою.

Таблиця 1. Варіанти завдання

Номер варіанту	1	2	3	4	5	6	7	8	9
Номер програми	1	3	2	4	2	1	4	3	2

3.5. Відтрансльуйте програмний проект.

3.6. Переконайтеся, що при трансляції програмного модуля не виявлені синтаксичні помилки.


3.7. Переконайтеся, що в директорії проекту створені завантажувальний файл з розширенням .lst і завантажувальний hex-файл з розширенням .hex.


3.8. Виконайте покрокове налагодження програми з використанням кнопки F11. На кожному кроці виконання програми запишіть значення використовуваних регістрів відповідно до таблиці 2.

Таблиця 2. Результати виконання команд


№	Команда	Код	Виконувана операція	Вміст використовуваних регістрів і комірок пам'яті до і після виконання		Пояснення
				До	Після	
1				A/00 PC/00 PSW/00	A/F2 PC/01 PSW/01	
2
...
...


4. Методологічні вказівки до виконання роботи

4.1. Увімкніть ПЕОМ, і викличте інтегроване середовище програмування, клацнувши мишею по значку .

4.2. Створіть новий файл вихідного тексту програми, клацнувши мишею по значку .

4.3. Введіть заданий текст програми, використовуючи клавіатуру.


4.4. Відкрийте діалогове вікно збереження файлу, клацнувши мишею по значку .

4.5. Створіть нову папку з ім'ям **LAB1**. Для цього клацніть мишею по значку .

4.6. Введіть ім'я файлу, наприклад **L1.asm** (розширення файлу повинна бути обов'язково) і натисніть на кнопку клавіатури **"Enter"**.

4.7. Створіть новий проект. Для цього виберіть підміню **"New project"** з меню **"project"**. Виберіть папку розміщення нового проекту - **LAB1**. Вкажіть ім'я проекту - **L1**.

4.8. Підключіть до програмного проекту файл з початковим текстом програми. Для цього можна натиснути правою кнопкою миші по значку групи файлів у вікні менеджера проектів і вибрати опцію додавання файлів до програмного проекту.

4.9. Відтрансляйте проект, натиснувши кнопку  (**Rebuild all target files**). Якщо є помилки, виправте їх в текстовому редакторі середовища Keil.

4.10. При відсутності помилок переконайтеся, що в директорії проекту з'явився завантажувальний файл з розширенням **.hex**.

4.11. Для виклику відлагоджувача натисніть піктограму його запуску .

4.12. Виконайте покрокову налагодження програми з використанням кнопки **F11**.

На кожному кроці виконання програми запишіть значення використовуваних змінних програми A [I] і S.

Якщо при виклику відлагоджувача в нижній частині екрана відсутнє вікно перегляду змінних, воно вмикається вибором в меню **View команду Watch & call stack window**.

Для додавання змінної у вікно **Watch** досить натиснути правою кнопкою миші по потрібній змінній у вікні програми і у вікні вибрати **ADD "ім'я змінної" to Watch window**.

5. Зміст звіту

5.1. Оригінальний текст програми.

5.2. Файл лістингу (файл з розширенням .lst).

5.3. Роздруківка завантажувального файлу (з розширенням .hex).

5.4. Таблиця значень зміни регістрів мікроконвертора на кожному кроці виконання за прикладом таблиці 2.